## I CLAIMS

1. A method of controlling execution of a processing task within a data processing system, said method comprising the steps of:

executing said processing task including allocating memory areas for data storage; and

suspending an actual execution path of said processing task at an execution point to perform memory management, said memory management comprising the steps of:

identifying one or more data items occurring in the course of execution and accessible to said processing task at said execution point which specify reference values pointing to respective ones of said memory areas;

determining a correlation between reference values corresponding to identified data items and memory areas allocated during said execution up to said execution point; and

performing a memory management operation on allocated memory areas in dependence upon said correlation.

2. A method as claimed in claim 1, wherein each of said one or more data items is an operand.

3. A method as claimed in claim 2, wherein said identifying step comprises:

identifying a possible execution path leading to said execution point, wherein said possible execution path may be different from said actual execution path;

performing a simulated execution of said possible execution path; and

wherein said one or more data items accessible to said processing task are identified by following said possible execution path to said current execution point.

4. A method as claimed in claim 3, wherein said memory management operation comprises marking all of said memory areas that are accessible to said processing task either directly or indirectly through said identified data items and collecting umarked memory areas for re-allocation during subsequent execution of said processing task.

5. A method as claimed in claim 4, wherein said memory management operation comprises compacting said unmarked memory areas prior to re-allocation.

6. A method as claimed in claim 1, wherein each of said one or more data items is a local variable.

7. A method as claimed in claim 6, wherein said identifying step comprises:

scanning a plurality of program instructions corresponding to said processing task and logging a data type for each store instruction corresponding to each of said one or more data items;

categorising at least one of said one or more data items as a multiple-type variable if different data types are logged for different store instructions for a respective data item;

simulating all possible execution paths up to said execution point for each of said one or more data items categorised as a multiple-type variable and determining the data type associated with each multiple-type variable at each of said plurality of program instructions for each of said possible execution paths; and

checking said determined data type for each of said multiple-type variables at one of said plurality of program instructions corresponding to said current execution point; and

said memory management operation is performed in dependence upon a result of said step of checking said determined data type.

8. A method as claimed in claim 7, wherein said memory management operation involves tagging said data item as suitable for reallocation if said determined data type

is different for different ones of said possible execution paths at said current execution point.

9. A method as claimed in claim 1, wherein said processing task is a component of a computer program written in an object-oriented programming language.

10. A method as claimed in claim 9, wherein said object-oriented programming language is Java.

11. A computer program product bearing a computer program for controlling a computer to control execution of a processing task within a data processing system, said computer program comprising:

execution code operable to execute said processing task including allocating memory areas for data storage; and

suspending code operable to suspend an actual execution path of said processing task at an execution point to perform memory management;

reference identifying code operable to identify one or more data items occurring in the course of execution and accessible to said processing task at said execution point which specify reference values pointing to respective ones of said memory areas;

correlating code operable to determine a correlation between reference values corresponding to identified data items and memory areas allocated during said execution up to said execution point; and

memory management code operable to perform a memory management operation on allocated memory areas in dependence upon said correlation.

12. A computer program product as claimed in claim 11, wherein each of said one or more data items is an operand.

13. A computer program product as claimed in claim 12, wherein said reference identifying code comprises:

path identifying code operable to identifying a possible execution path leading to said execution point, wherein said possible execution path may be different from said actual execution path;

path simulation code operable to perform a simulated execution of said possible execution path; and

wherein said one or more data items accessible to said processing task are identified by following said possible execution path to said current execution point.

14. A computer program product as claimed in claim 13, wherein said memory management code is operable to mark all of said memory areas that are accessible to said processing task either directly or indirectly through said identified data items and to collect umarked memory areas for re-allocation during subsequent execution of said processing task.

15. A computer program product as claimed in claim 14, wherein said memory management code is operable to compact said unmarked memory areas prior to re-allocation.

16. A computer program product as claimed in claim 11, wherein each of said one or more data items is a local variable.

17. A computer program product as claimed in claim 16, wherein said reference identifying code comprises:

scanning code operable to scan a plurality of program instructions corresponding to said processing task and to log a data type for each store instruction corresponding to each of said one or more data items;

categorising code operable to categorise at least one of said one or more data items as a multiple-type variable if different data types are logged for different store instructions for a respective data item;

path simulation code operable to simulate all possible execution paths up to said execution point for each of said one or more data items categorised as a multiple-type variable and to determine the data type associated with each multiple-type variable at each of said plurality of program instructions for each of said possible execution paths; and

data type checking code operable to check said determined data type for each of said multiple-type variables at one of said plurality of program instructions corresponding to said current execution point; and

wherein said memory management code is operable to perform said memory management operation in dependence upon a result of said data type checking code.

18. A computer program product as claimed in claim 17, wherein said memory management code is operable to tag said data item as suitable for reallocation if said determined data type is different for different ones of said possible execution paths at said current execution point.

19. A computer program product as claimed in claim 1, wherein said processing task is a component of a computer program written in an object-oriented programming language.

20. A computer program product as claimed in claim 19, wherein said object-oriented programming language is Java.

21. A data processing apparatus operable to control execution of a processing task within a data processing system, said data processing apparatus comprising:

execution logic operable to execute said processing task including allocating memory areas for data storage; and

task suspension logic operable to suspend an actual execution path of said processing task at an execution point to perform memory management;

reference identifying logic operable to identify one or more data items occurring in the course of execution and accessible to said processing task at said

execution point which specify reference values pointing to respective ones of said memory areas;

correlation logic operable to determine a correlation between reference values corresponding to identified data items and memory areas allocated during said execution up to said execution point; and

memory management logic operable to perform a memory management operation on allocated memory areas in dependence upon said correlation.

22. A data processing apparatus as claimed in claim 21, wherein each of said one or more data items is an operand.

23. A data processing apparatus as claimed in claim 22, wherein said reference identifying logic comprises:

path identifying logic operable to identify a possible execution path leading to said execution point, wherein said possible execution path may be different from said actual execution path;

path simulation logic operable to perform a simulated execution of said possible execution path; and

wherein said one or more data items accessible to said processing task are identified by following said possible execution path to said current execution point.

24. A data processing apparatus as claimed in claim 23, wherein said memory management logic is operable to mark all of said memory areas that are accessible to said processing task either directly or indirectly through said identified data items and collecting umarked memory areas for re-allocation during subsequent execution of said processing task.

25. A data processing apparatus as claimed in claim 24, wherein said memory management logic is operable to compact said unmarked memory areas prior to re-allocation.

26. A data processing apparatus as claimed in claim 21, wherein each of said one or more data items is a local variable.

27. A data processing apparatus as claimed in claim 26, wherein said reference identifying logic comprises:

scanning logic operable to scan a plurality of program instructions corresponding to said processing task and logging a data type for each store instruction corresponding to each of said one or more data items;

categorising logic operable to categorise at least one of said one or more data items as a multiple-type variable if different data types are logged for different store instructions for a respective data item;

path simulation logic operable to simulate all possible execution paths up to said execution point for each of said one or more data items categorised as a multiple-type variable and to determine the data type associated with each multiple-type variable at each of said plurality of program instructions for each of said possible execution paths; and

data type checking logic operable to check said determined data type for each of said multiple-type variables at one of said plurality of program instructions corresponding to said current execution point; and

said wherein memory management logic is operable to perform said memory management operation in dependence upon a result of checking said determined data type.

28. A data processing apparatus as claimed in claim 27, wherein said memory management logic is operable to tag said data item as suitable for reallocation if said determined data type is different for different ones of said possible execution paths at said current execution point.

29. A data processing apparatus as claimed in claim 1, wherein said processing task is a component of a computer program written in an object-oriented programming language.

30. A data processing apparatus as claimed in claim 29, wherein said object-oriented programming language is Java.